SUNMOS

Rolf Riesen, Lee Ann Fisk, Chu Jong, Barney Maccabe,
Kevin McCurley, Lance Shuler, Mack Stallcup, David van Dresser

1 What is SUNMOS?

SUNMOS (Sandia/UNM Operating System) is a joint project between Sandia National Laboratories and the computer science department at the University of New Mexico (UNM)[1]. The goal of the SUNMOS project is to develop a highly portable, yet efficient, operating system for massively parallel distributed memory systems. While the project was initially driven by the need to support scientific applications, we have broadened our interests to include business applications, specifically applications that process extremely large numbers of transactions.

The project was initiated in January 1991. The first step was to develop an operating system that was compatible with Vertex, the vendor supplied operating system for the nCUBE-2. SUNMOS, the result of this effort, has served as a test-bed for new message passing schemes and exploration into high performance I/O.

In Spring 1993 we decided to port SUNMOS to the Intel Paragon. It took only three month to do so, attributing to the portability of SUNMOS. Efforts are currently directed at making SUNMOS more robust and adding features that have been identified as necessary by our growing user community.

Especially at the beginning, when the discrepancies between SUNMOS and OSF were much larger, some people thought it would be funny to compare[2] the two operating systems.

Puma[3] Is the successor of SUNMOS.

2 How does SUNMOS work?

SUNMOS is not a full blown operating system. It is targeted at massively parallel systems. Such systems are currently termed "supercomputers" and priced accordingly. The goal of SUNMOS is to make as much of the hardware as possible available to the application programmer. Where we have to chose between performance and added functionality, the decision usually favors performance. The idea is, that if the functionality is really needed, the application or library can provide that without degrading performance for other users and applications.

Therefore, we try to keep SUNMOS as slim as possible and provide only the functionality needed by all applications. Not surprisingly then, most of SUNMOS is concerned with reliable and fast message passing. The kernel image is less than 32 kB for the nCUBE 2,

[1] http://www.cs.unm.edu/
[2] http://www.swcp.com/%7Emccurley/humor/sunmos_humor.html
[3] http://www.cs.sandia.gov/puma

and consumes less than 256 kB of memory on an Intel Paragon node. This compares very favorably to the several mega bytes occupied by the Mach kernel and OSF/1 AD.

SUNMOS is intended to take over the compute nodes of a system. Applications are started and controlled from a host node process called yod. The yod program runs on a SUN frontend for the nCUBE 2, and on a service node on the Intel Paragon. Yod can make use of all services provided by the host operating system, and make them available to the application running on the SUNMOS nodes.

SUNMOS is a single-tasking kernel. (Puma offers the capability to run multiple tasks on the same node.) SUNMOS does not provide demand paging. We feel that the overhead of several thousand nodes accessing the same page, for example during program load, is not justified for our intended target machines. While a parallel application under OSF may take several minutes to load and page out unnecessary parts of the operating system, it never takes more than a few seconds to load a SUNMOS application. No matter how big and onto how many nodes.

Once a SUNMOS application is loaded and running, it can manage practically all of the available memory on a node and use the full resources provided by the hardware. Requests such as standard I/O are directed to yod which uses the host OS to grant them.

3 SUNMOS performance

So, given the limited features and the aim to keep it small and lean, what do we get in return? We have already mentioned memory on Intel Paragon nodes. While OSF and Mach easily take up five mega bytes of memory, SUNMOS uses less than 256 kB out of the available 16 MB or 32 MB. The network in our machine is rated at 175 MB/s. SUNMOS applications can transmit messages at up to 160 MB/s, while OSF applications have to content themselves with about 65 MB/s (OSF 1/AD R1.2).

User to user latencies, starting with OSF 1/AD R1.2, are now well below 40 micro seconds for OSF (using the message co-processor). However, due to paging and other issues, this is only a lower bound. Tests reveal widely fluctuating numbers. Zero-length messages under SUNMOS have a latency of less than 25 micro seconds. Using the second i860 on a SUNMOS node as a message co-processor (with the -proc 1 option to yod), drops that number below the 20 micro seconds mark. One of the advantages of SUNMOS is its predictability. There is no paging or task switching that could mysteriously interfere with bandwidth or latency. The following two graphs illustrate this.

The error bars show the difference between the maximum and minimum value measured. The curve is the average of 1000 trials. The timing for each trial starts at the time when the user program on node A executes the call to send the data, and ends when the user process on node B has access to all the data. Instead of explaining the procedure in detail, have a look at the source code for tsmsg.c . The program uses hardware broadcast to synchronize two nodes which then exchange messages.

Unfortunately, hardware broadcast can dead-lock. This means exclusive access to the machine is required. Furthermore, hardware broadcast is not available under OSF. Results obtained using this test cannot be compared to other benchmarks under OSF.

All timings where done using step-B NICs and running the back plane at 175 MB/s (slow streaming mode).

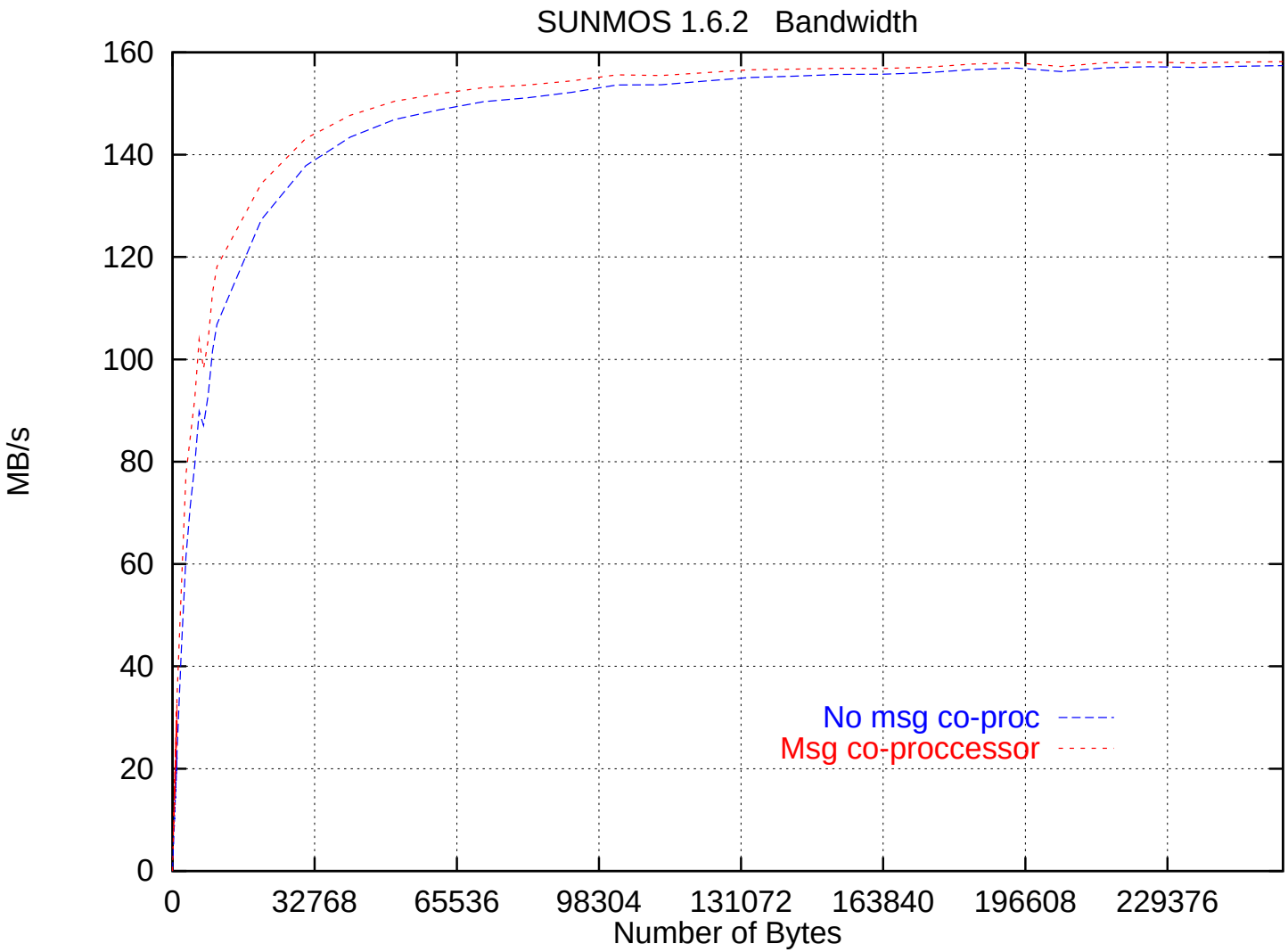

Figure 1: Bandwidth

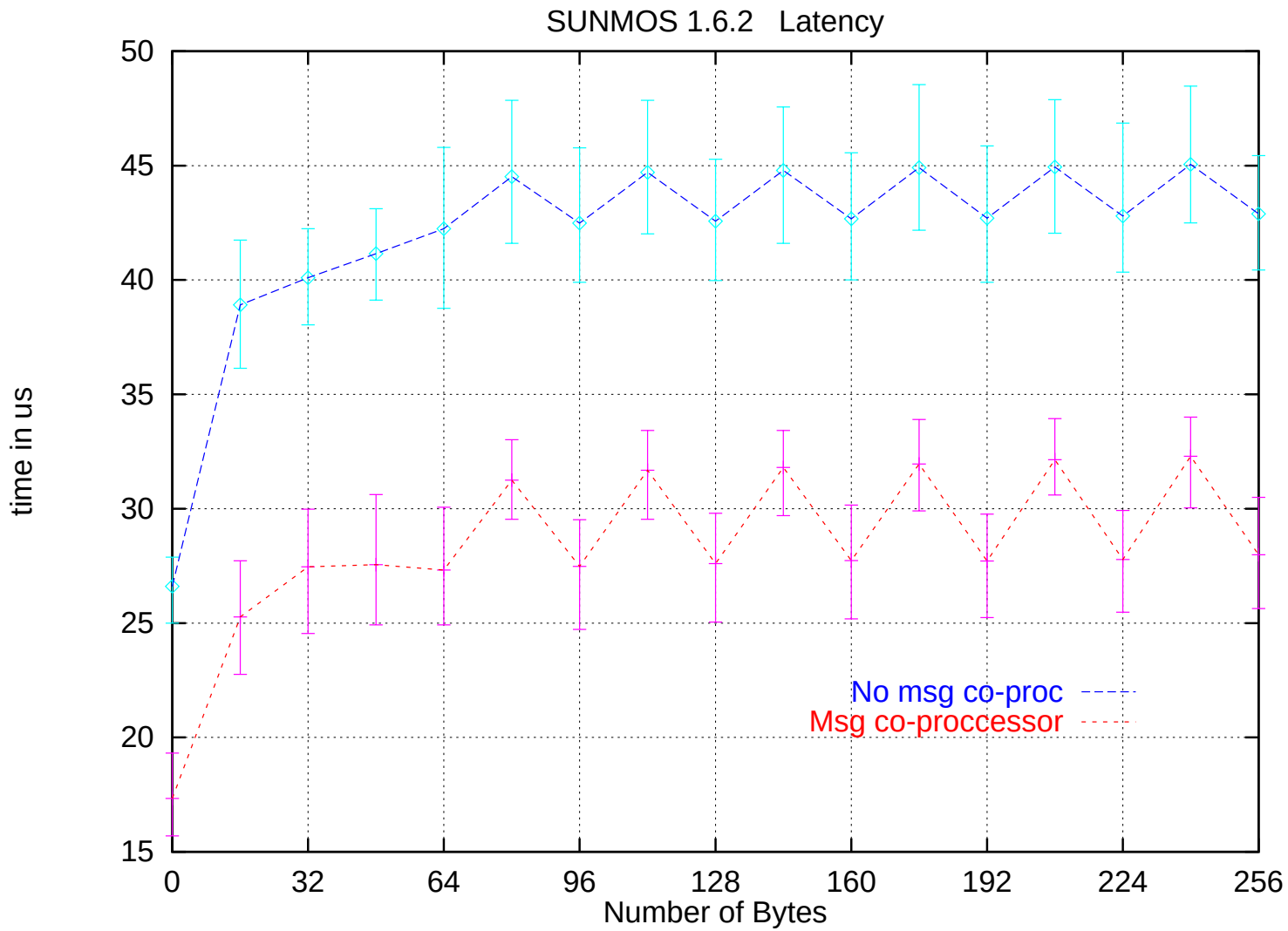

Figure 2: Latency

The Intel comsat test that is part of the acceptance test suit runs under OSF and SUNMOS (with some limitations). Here are the results for comparison.

```
-------------------------------------------------
SUNMOS 1.6.3 results without coprocessor:

Unforced CSEND Latency (microseconds): 19
Forced   CSEND Latency (microseconds): 18
Unforced ISEND Latency (microseconds): 28
Forced   ISEND Latency (microseconds): 26

Stats for Standard Latency, alpha (microseconds):

Maximum = 33
Minimum = 29
Mean    = 29
Median  = 29
Std_dev = 0


-------------------------------------------------
SUNMOS 1.6.3 results with coprocessor:

Unforced CSEND Latency (microseconds): 17
Forced   CSEND Latency (microseconds): 16
Unforced ISEND Latency (microseconds): 23
Forced   ISEND Latency (microseconds): 21

Stats for Standard Latency, alpha (microseconds):

Maximum = 27
Minimum = 25
Mean    = 25
Median  = 25
Std_dev = 0


-------------------------------------------------
OSF R1.2.6 results with coprocessor:

Unforced CSEND Latency (microseconds): 29
Forced   CSEND Latency (microseconds): 151
Unforced ISEND Latency (microseconds): 16
Forced   ISEND Latency (microseconds): 22

Stats for Standard Latency, alpha (microseconds):
```

```
Maximum = 15804
Minimum = 45
Mean    = 678
Median  = 46
Std_dev = 3087
```

Bear in mind, that the measurements for SUNMOS include some overhead for the NX compatibility library.

SUNMOS makes it also possible to use the second CPU (and third CPU on a MP node) as a compute processor. It is this feature that has enabled us to get more Gflops out of our Paragon, than what it was rated for. It also allowed us to reclaim the world MP-LINPACK record[4] [1].

4 Q & A, Hints and Tips

In this section I would like to collect answers to commonly asked questions, as well as hints and tips on SUNMOS usage. Something along the lines of David Robboy's article[5] he wrote for Intel On-Line[6].

Another example is Ted Barragy's article[7].

5 Documentation

For an introduction on how to use SUNMOS on the Intel Paragon, see [4]. The paper[8] is available in Postscript format and can be down loaded.

There are two documents that help you install SUNMOS. The SUNMOS Installation Guide[9] takes you step by step through the process. If you want to install single node[10] boot capability, this document might be of interest to you.

A document describing the NX emulation[11] under SUNMOS is also available. It contains a list of NX calls that are fully supported, partially supported, or not implemented in SUNMOS.

There is also a document that describes the use of active messages[12] under SUNMOS. It is also available in postscript[13] format.

There are several man pages. System administrators might be interested in create_yod_config[14]. This program is used, usually at boot time, to setup the configuration file

[4] http://www.ssd.intel.com/press/record.html
[5] http://www.ssd.intel.com/cgi-bin/InfoSel/get-infoselect.pl?30009
[6] http://www.ssd.intel.com/IOL/index.html
[7] http://www.ssd.intel.com/cgi-bin/InfoSel/get-infoselect.pl?30021
[8] file://www.cs.sandia.gov/pub/sunmos/papers/ISUG94-1.ps.Z
[9] file://www.cs.sandia.gov/pub/sunmos/doc/info/sunmos.install.ps
[10] file://www.cs.sandia.gov/pub/sunmos/doc/info/single_boot.ps
[11] file://www.cs.sandia.gov/pub/sunmos/doc/info/nxemulation.ps
[12] http://www.cs.sandia.gov/%7Erolf/am/am.html
[13] http://www.cs.sandia.gov/%7Erolf/misc/am.ps
[14] file://www.cs.sandia.gov/pub/sunmos/doc/man/create_yod_config.l.html

SUNMOS needs to keep track of the currently running jobs. Fyod[15] is a servers that allows parallel access to the file system. The companion program fservers[16] lists the directories that are currently managed by fyod. The program release_job[17] can be used to removed entries in the SUNMOS configuration file.

Users should read the man pages for yod[18], showparts[19], showmesh[20], nsend[21], and groups[22]. The `yod` program is used to load and control user programs. It has (too) many options, and this document explains them all. `Showparts` and `showmesh` are used to display the currently running jobs. `Showmesh` only lists SUNMOS jobs, while `Showparts` also shows OSF jobs. The man page for `nsend` explains how to use the lowest level message passing mechanisms available under SUNMOS. If you want to load and run heterogeneous applications, you should have a look at the `groups` document.

To help debug SUNMOS programs, you should have a look at the getpcb[23], and getcomm[24]. man pages. The usage of these programs and how to interpret their output is explained in the SUNMOS User Guide[25].

These man pages are also available in troff format[26].

6 Papers

At the end of this document you will find a list of papers, articles, and techreports. To be included in the list the paper must be somehow related to SUNMOS (or Puma). Let me know if you know of one that should be there, but is not. The list[27] is available in BibTeX format.

Several papers talk about results using SUNMOS on the Intel Paragon. [2] and [3] were presented at ISUG-93, shortly after we completed the port of SUNMOS to the Paragon. The first high Gflops number under SUNMOS is described in [17]. The two Gordon Bell award entries that were using SUNMOS are described in [10] and [15]. SUNMOS on the Intel Paragon is compared to an SP1 and the T3D using the shallow water model in [19]. Raw numbers are reported in [1]. Since breaking the world record[28], SUNMOS leads the list!

There are two papers comparing SUNMOS and OSF 1/AD on the Intel Paragon: [12] and [9].

We have been, and will again, look at issues in parallel I/O. Most of this work was done under SUNMOS on the nCUBE 2. We implemented a parallel file system and measured its

[15] file://www.cs.sandia.gov/pub/sunmos/doc/man/fyod.l.html
[16] file://www.cs.sandia.gov/pub/sunmos/doc/man/fservers.l.html
[17] file://www.cs.sandia.gov/pub/sunmos/doc/man/release_job.l.html
[18] file://www.cs.sandia.gov/pub/sunmos/doc/man/yod.l.html
[19] file://www.cs.sandia.gov/pub/sunmos/doc/man/showparts.l.html
[20] file://www.cs.sandia.gov/pub/sunmos/doc/man/showmesh.l.html
[21] file://www.cs.sandia.gov/pub/sunmos/doc/man/nsend.l.html
[22] file://www.cs.sandia.gov/pub/sunmos/doc/man/groups.l.html
[23] file://www.cs.sandia.gov/pub/sunmos/doc/man/getpcb.l.html
[24] file://www.cs.sandia.gov/pub/sunmos/doc/man/getcomm.l.html
[25] file://www.cs.sandia.gov/pub/sunmos/papers/ISUG94-1.ps.Z
[26] file://www.cs.sandia.gov/pub/sunmos/doc/man
[27] file://www.cs.sandia.gov/pub/sunmos/papers/puma.bib
[28] http://www.ssd.intel.com/press/record.html

performance in [6]. Reading and writing large matrices from secondary story is dealt with in [16] and [18].

We also looked at active messages and ported Split-C to SUNMOS. Our implementation and performance numbers are described in [7] and [8].

7 How to Get SUNMOS for the Intel Paragon?

In general, due to Sandia policies and US government export laws, we can not let you have the source code to SUNMOS. There are ways to make it possible despite that. However, very few sites have acquired a source license so far. Contact Rolf Riesen if you want to learn more.

We are able to provide you with binaries, even outside the US. There are still some restrictions and we ask that you sign a license agreement. The license is mostly to protect us from law suits and should also prevent you from giving SUNMOS to other sites. The license is free for US government laboratories and Universities. Commercial sites have to pay a license fee. Get in touch with Rolf Riesen for a license. Once that has been arranged, Mack Stallcup will interface with your Intel PSE to get it installed.

There is a README[29] file that gives information about the latest release.

8 SUNMOS Support

Currently, the people who developed SUNMOS are also providing support for it. We answer questions, fix bugs, and add some enhancements. When we have collected enough changes or fixed some major bugs that prevent many users from running their programs, then we gather everything together and create a new release. The latest release is S1.6.6[30]. (Also, see the latest README[31] file for that release.)

Since we are such a small team, it may sometimes take a while for a bug to get fixed, or a feature to be added. We prioritize by the number (and importance) of users impacted.

Intel SSD now fully supports the interface between OSF 1/AD and SUNMOS. Most people there know about SUNMOS/Puma, and at least one (David Robboy robboy@SSD.intel.com), is a SUNMOS expert. David has SUNMOS running on several Paragon's at SSD and has helped us and many users with SUNMOS/OSF issues.

9 Who is Using SUNMOS?

The following sites have acquired SUNMOS. We have no information whether they are still using it and how much, if so. Most sites seem to run SUNMOS on some nodes, and OSF on the others. We would like to gather statistics about SUNMOS vs. OSF usage. If any of the SUNMOS sites are on the WEB, please send me your URL, so I can add a hyper link from here to your site.

- Boeing

[29] file://www.cs.sandia.gov/pub/sunmos/distrib/README.latest_release
[30] file://www.cs.sandia.gov/pub/sunmos/distrib/S1.6.5
[31] file://www.cs.sandia.gov/pub/sunmos/distrib/README.latest_release

- California Institute of Technology
- Carnegie Mellon University
- DOE Ames lab, Iowa
- Graz University of Technology, Austria
- Honeywell Space Systems, Clearwater, Florida
- Indiana University
- Intel SSD
- Intel SSD, United Kingdom
- IRISA/INRIA, France
- NASA Ames
- NASA Langley
- Naval Air Warfare Center, China Lake
- NOAA Forecast Systems Laboratory
- NRaD/NCCOSC, San Diego
- NSA
- Oak Ridge National Labs
- Purdue University
- San Diego Supercomputer Center (SDSC)
- SUNY at Stony Brook
- University of Arizona
- University of Bergen, Norway First non-US site!
- University of Puerto Rico
- University of Technology, RWTH Aachen, Germany[32]
- University of South Carolina
- Wright-Patterson AFB, OH

[32] http://www.lfbs.rwth-aachen.de

10 About the SUNMOS/Puma Mailing List

Currently, more than one hundred people from over thirty organizations are on this list (universities, research labs, and vendors). The mailing list is open for discussions of all topics concerning SUNMOS and Puma. The topic list includes but is not limited to:

- PUMA the successor to SUNMOS.
- The future of Operating Systems for MP machines.
- SUNMOS/PUMA on various machines (Paragon, nCUBE 2, ???).
- Questions and suggestions for SUNMOS/PUMA.
- User experiences with SUNMOS.
- Success stories using SUNMOS.
- Trouble using SUNMOS.
- Work arounds for SUNMOS peculiarities.
- etc.

Send e-mail to rolf@cs.sandia.gov to subscribe.

11 The People Behind SUNMOS and Puma

The following people are currently involved in the development of SUNMOS and Puma at Sandia:

David van Dresser (Sandia) Joined in 1992 and wrote most of the kernel for Puma on the nCUBE 2. He is currently involved in writing the kernel for the Intel Paragon. *dwvandr@cs.sandia.gov*

T. Mack Stallcup (Intel) is one of the on-site PSEs at Sandia. He was a member of the SUNMOS team from the start, and is currently responsible for the distribution of SUNMOS to other Intel Paragon sites. *tmstall@cs.sandia.gov*

Lance Shuler (Sandia) joined the group in 1994. He has researched global and group collective communication paradigms and is currently writing libraries to make efficient use of portals. *shuler@cs.sandia.gov*

Rolf Riesen (Sandia) joined the group in 1992 and has now taken over the management of the project at Sandia. *rolf@cs.sandia.gov*[33]

Kevin McCurley (Sandia) is not really a member of the SUNMOS team. However, he got tired of the lacking support of Intel NX calls, such as isend/irecv, gcol, etc. Therefore, he wrote and maintains the Intel compatibility libraries. *Kevin*[34]

[33] http://www.cs.unm.edu:80/%7Eriesen/
[34] http://www.swcp.com/%7Emccurley/index.html

Chu Jong (Sandia) joined the team in 1994. His currently investigating options for debugger support under Puma. *cjong@cs.sandia.gov*

Lee Ann Fisk (Sandia) joined us as a research assistant in 1994 and is now a Sandia employee. *lafisk@cs.sandia.gov*

At UNM, Barney Maccabe and his students form the second part of our group:

Arthur B. Maccabe (UNM) is co-founder (with Stephen Wheat) of the project and responsible for the involvement of UNM. *maccabe@cs.unm.edu*[35]

Brian Sanchez (UNM) is currently involved in the unified[36] project. *bsanchez@cs.unm.edu*

J. Lance Mumma (UNM) is writing collective communications libraries. *mumma@cs.sandia.gov*

Betsy Matthews (UNM) is investigating memory allocation issues in the PCT. *betsy@cs.sandia.gov*

During the Summer we usually have additional people helping us. This is no different this year: Ron Brightwell (*rbbrigh@cs.sandia.gov*) from Mississippi State is porting MPI on top of Puma portals, Heather Richards[37] from Louisiana Tech is working on a showmesh for Puma, Steve Chapin (*sjc@mcs.kent.edu*) from Kent State is involved in the unified kernel project, and Tramm Hudson[38] from Tulane University is implementing a collective communications library.

Over the years, many people have been involved in the project and contributed. We wish to acknowledge the following people: Al Audette, Miguel Alvarez, Anthony Ferrara, Kenneth Ingham, Gabi Istrail, Clint Kaul, Lisa Kennicott, Michael Levenhagen, Francisco Reverbel, Doug Sanchez, Judy Sturtevant, Bob van Sant, Jeff VanDyke.

Last, but not least, we would like to achnowlege the founder of this effort. Stephen Wheat *srwheat@ssd.intel.com* has left Sandia and is now working as a software manager for Intel SSD. He heads the leight weight kernel (LWK) effort.

12 E-Mail Addresses and FTP Site

sunmos-support@cs.unm.edu Bug reports and questions should be sent here. Note, this mail alias is at UNM.

sunmos-dev@cs.sandia.gov Requests for enhancements, bug fixes, suggestions, and questions addressed to the developers of SUNMOS should be sent here.

puma-dev@cs.sandia.gov Mail to the Puma developers should go here.

sunmos-news@cs.sandia.gov The SUNMOS/Puma mailing list. Questions and comments of general interest to the SUNMOS community should be sent here. Even though most of the current SUNMOS activity is centered around the Intel Paragon, this mailing list is not for Paragon user's alone. SUNMOS currently runs on the

[35] http://www.cs.unm.edu/ maccabe.html
[36] http://www.cs.sandia.gov/%7Erolf/puma/unified/unified.html
[37] http://www.latech.edu/%7Ehcr/
[38] http://www.cs.tulane.edu/ hudson/

nCUBE 2, and ports to other vendors and future generations of current machines are under way. General discussions of OS related topics for massively parallel distributed memory architectures are also welcome. (You have to be a member of the mailing list in order to send to this address. See next item.)

rolf@cs.sandia.gov Maintainer of the sunmos-news mailing list. Send requests to be added or dropped to this address.

ftp.cs.sandia.gov Anonymous ftp site for SUNMOS and Puma. The following directories might be of interest:

- /pub/sunmos/doc/man[39]
- /pub/sunmos/doc/info[40]
- /pub/sunmos/papers[41]

13 Other Interesting Sites

The Paragon manuals[42] are available on-line. Also at CalTech is a page that has pointers to other Supercomputing Servers[43]. A Parallel list[44] with pointers to online documents is located at CalTech as well.

David Bader maintains an extensive list of pointers to other Parallel Sites[45].

Intel SSD[46] is present on the WEB, as well as the Intel Supercomputer Users' Group[47].

Sandia National Laboratories[48] has its own page, and so does the MPCRL[49].

A table of contents, abstracts, and some of the papers of Supercomputing'94[50] are accessible through the WEB.

If you want to know who currently owns the most computing power, consult the List of the world's most powerful computing sites[51]. It is updated weekly.

The book Designing and Building Parallel Programs[52], by Ian Foster is available online in its entirety.

David Kotz maintains the Parallel I/O Archive at Dartmouth[53]. It is a very good starting point to find out what is going on in parallel I/O research.

If you want to learn more about Paradyn[54]'s performance tools, checkout this site.

[39] file://www.cs.sandia.gov/pub/sunmos/doc/man
[40] file://www.cs.sandia.gov/pub/sunmos/doc/info
[41] file://www.cs.sandia.gov/pub/sunmos/papers
[42] http://www.ccsf.caltech.edu/paragon/man.html
[43] http://www.ccsf.caltech.edu/other_sites.html
[44] http://www.ccsf.caltech.edu/documentation.html
[45] http://www.umiacs.umd.edu/%7Edbader/sites.html
[46] http://www.ssd.intel.com/homepage.html
[47] http://www.ssd.intel.com/ISUG/isug-home.html
[48] http://www.sandia.gov/
[49] http://www.cs.sandia.gov/
[50] http://www.computer.org/p3/sc94home.html
[51] http://yarrow.wt.uwa.edu.au:80/%7Egunter/
[52] http://www.mcs.anl.gov/dbpp
[53] http://www.cs.dartmouth.edu/pario.html
[54] http://www.cs.wisc.edu/%7Eparadyn/

The description of our MP Linpack world record[55] includes pictures and more detail than the press release.

If you know of a pointer to a site that should be included here, please send mail to rolf@cs.sandia.gov.

References

[1] Jack J. Dongarra. Performance of various computers using standard linear equations software. Technical report CS-89-85, Computer Science Department, University of Tennessee and Mathematical Sciences Section, Oak Ridge National Laboratories, January 1995. Available from: file://netlib.att.com/netlib/benchmark/performance.ps.Z .

[2] David Greenberg, Barney Maccabe, Kevin S. McCurley, Rolf Riesen, and Stephen Wheat. Communication on the Paragon. In *Proceedings of the Intel Supercomputer Users' Group. 1993 Annual North America Users' Conference.*, pages 117–124, October 1993. Available from: file://www.cs.sandia.gov/pub/papers/mccurley/stats.ps .

[3] David S. Greenberg, Barney Maccabe, Rolf Riesen, Stephen Wheat, and David Womble. Achieving high performance on the Intel Paragon. In *Proceedings of the Intel Supercomputer Users' Group. 1993 Annual North America Users' Conference.*, pages 203–208, October 1993.

[4] Arthur B. Maccabe, Kevin S. McCurley, Rolf Riesen, and Stephen R. Wheat. SUNMOS for the Intel Paragon: A brief user's guide. In *Proceedings of the Intel Supercomputer Users' Group. 1994 Annual North America Users' Conference.*, pages 245–251, June 1994. Available from: file://www.cs.sandia.gov/pub/sunmos/papers/ISUG94-1.ps.Z .

[5] Arthur B. Maccabe and Stephen R. Wheat. Message passing in PUMA. Technical report SAND93-0935, Sandia National Laboratories, 1993. Available from: file://www.cs.sandia.gov/pub/sunmos/papers/sand.93-0935.ps.Z .

[6] Rolf Riesen, Arthur B. Maccabe, and Stephen R. Wheat. Experience in implementing a parallel file system. Unpublished, 1993. Available from: file://www.cs.sandia.gov/pub/sunmos/papers/pfs.ps.Z .

[7] Rolf Riesen, Arthur B. Maccabe, and Stephen R. Wheat. Active messages versus explicit message passing under SUNMOS. In *Proceedings of the Intel Supercomputer Users' Group. 1994 Annual North America Users' Conference.*, pages 297–303, June 1994. Available from: file://www.cs.sandia.gov/pub/sunmos/papers/ISUG94-2.ps.Z .

[8] Rolf Riesen, Arthur B. Maccabe, and Stephen R. Wheat. Split-C and active messages under SUNMOS on the Intel Paragon. unpublished, 1994.

[55] http://www.ssd.intel.com/press/record.html

[9] Subhash Saini and Horst D. Simon. Applications performance under OSF/1 AD and SUNMOS on Intel Paragon XP/2-15. In *Proceedings of Supercomputing'94*, pages 580–589, Washington, D.C., November 1994. Abstract available from: http://www.computer.org/p3/saini.html .

[10] John Shadid, Scott Hutchinson, Harry Moffat, Gary Hennigan, Bruce Hendrickson, and Robert Leland. A 65+ Gflop/s unstructured finite element simulation of chemically reacting flows on the Intel Paragon. In *Proceedings of Supercomputing'94*, pages 673–678, Washington, D.C., November 1994. Abstract available from: http://www.computer.org/p3/shadid.html .

[11] Lance Shuler, Rolf Riesen, Chu Jong, David van Dresser, Arthur B. Maccabe, Lee Ann Fisk, and T. Mack Stallcup. The Puma operating system for massively parallel computers. In *Proceedings of the Intel Supercomputer Users' Group. 1995 Annual North America Users' Conference.*, June 1995. Available from: file://www.cs.sandia.gov/pub/sunmos/papers/puma_isug95.ps.Z .

[12] Bernard Traversat, Bill Nitzberg, and Sam Fineberg. Experience with SUNMOS on the Paragon XP/S-15. In *Proceedings of the Intel Supercomputer Users' Group. 1994 Annual North America Users' Conference.*, pages 252–259, June 1994.

[13] Stephen R. Wheat, Arthur B. Maccabe, Rolf Riesen, David W. van Dresser, and T. Mack Stallcup. PUMA: An operating system for massively parallel systems. In *Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences*, pages 56–65. IEEE Computer Society Press, 1994. Available from: file://www.cs.sandia.gov/pub/sunmos/papers/hicss.ps.Z .

[14] Stephen R. Wheat, Arthur B. Maccabe, Rolf Riesen, David W. van Dresser, and T. Mack Stallcup. PUMA: An operating system for massively parallel systems. *Scientific Programming*, 3:275–288, 1994.

[15] David Womble, David Greenberg, Stephen Wheat, Robert Benner, Marc Ingber, Greg Henry, and Satya Gutpa. Applications of boundary element methods on the Intel Paragon. In *Proceedings of Supercomputing'94*, pages 680–684, Washington, D.C., November 1994. Paper available from: ftp://www.cs.sandia.gov/pub/papers/dewombl/Gordon_Bell_94.ps.Z .

[16] David Womble, David Greenberg, Stephen Wheat, and Rolf Riesen. Beyond core: Making parallel computer I/O practical. In *DAGS'93 Proceedings*, pages 56–63, June 1993. Available from: file://www.cs.sandia.gov/pub/papers/dewombl/parallel_io_dags93.ps.Z .

[17] David Womble, David Greenberg, Stephen Wheat, and Rolf Riesen. LU factorization and the LINPACK benchmark on the Intel Paragon. Technical report SAND94-0428, Sandia National Laboratories, 1994. Available from: file://www.cs.sandia.gov/pub/papers/dewombl/paragon_linpack_benchmark.ps.Z .

[18] David E. Womble, David S. Greenberg, Rolf E. Riesen, and Stephen R. Wheat. Out of core, out of mind: Practical parallel I/O. In *Proceedings of the Scalable Libraries*

Conference, pages 10–16. Mississippi State University, October 1993. Available from: file://www.cs.sandia.gov/pub/papers/dewombl/parallel_io_scl93.ps.Z .

[19] Patrick H. Worley, Ian T. Foster, and Brian Toonen. Algorithm comparison and benchmarking using a parallel spectral transform shallow water model. In *Proceedings of the Sixth Workshop on Parallel Processing in Meteorology*, Singapore, 1994 (to appear). World Scientific. Available from: http://www.epm.ornl.gov/%7Eworley/papers/ecmwf94.ps .